What is claimed is:

1. A system for performing kernel-mode operations comprising:

a kernel-mode interface generator for generating a kernel-mode interface driver, which in turn generates a call gate to perform a kernel-mode operation with kernel-mode authorization in a kernel mode; and

an authorization interface, coupled to the kernel-mode interface generator, to connect a user mode to kernel mode, switching a process from user mode to kernel mode via the call gate to perform the kernel-mode operation.

2. The system as claimed in claim 1, wherein the authorization interface sends a call gate request to the kernel-mode interface generator to generate the kernel-mode interface driver, the call gate generated accordingly, and the authorization interface instructing the process to enter the kernel mode through the call gate.

3. The system as claimed in claim 1, wherein the kernel-mode operation is an operation with Ring 0 authorization level in the kernel mode.

4. The system as claimed in claim 1, wherein the process, a user-mode operation, is capable of user-mode authorization in a protected mode.

5. The system as claimed in claim 4, wherein the user-mode authorization is Ring 3 authorization level in the protected mode.

6. The system as claimed in claim 4, wherein the call gate sets a call gate selector and an entry point in a global descriptor table, having a call gate descriptor and a code-segment descriptor, to enable the process to perform the operation with kernel-mode authorization in the kernel mode.

7. The system as claimed in claim 6, wherein the user-mode authorization of the process is switched to kernel-mode authorization by the call gate selector via the entry point in the global descriptor table, and is switched back after the operation with kernel-mode authorization has been performed.

8. The system as claimed in claim 7, wherein a far call stated by the call gate selector points to the call gate descriptor, and a CPU switches an instruction pointer to the entry point, when a caller from the call gate gives a call, if the caller has kernel-mode authorization.

9. The system as claimed in claim 8, wherein the instruction pointer has kernel-mode authorization, is switched to the entry point, to perform the operation with kernel-mode authorization in the kernel mode, and is switched back to the user-mode authorization after the

operation with kernel-mode authorization has been performed.

10. A method for performing kernel-mode operations comprising steps of:

providing a kernel-mode generator;

generating a kernel-mode interface using the kernel-mode generator to generate a call gate performing a kernel-mode operation with kernel-mode authorization in a kernel mode;

providing an authorization interface to connect a user mode to the kernel mode; and

switching a process from the user mode to the kernel mode via the call gate through the authorization interface to perform the kernel-mode operation with kernel-mode authorization.

11. The method as claimed in claim 10, further comprising, in the step of providing the authorization interface:

sending of a call gate request by the authorization interface to the kernel-mode interface generator to generate the kernel-mode interface driver;

generating the call gate using the kernel-mode interface driver; and

the authorization interface instructing the process to enter the kernel mode through the call gate.

12. The method as claimed in claim 10, wherein in the generating step, the kernel-mode operation is an

operation with Ring 0 authorization level in the kernel mode.

13. The method as claimed in claim 10, wherein in the switching step, the process, a user-mode operation, is capable of user-mode authorization in a protected mode.

14. The method as claimed in claim 13, wherein the user-mode authorization is Ring 3 authorization level in the protected mode.

15. The method as claimed in claim 13, wherein in the generating step, the call gate sets a call gate selector and an entry point in a global descriptor table, having a call gate descriptor and a code-segment descriptor, to perform the operation with kernel-mode authorization in the kernel mode.

16. The method as claimed in claim 14, wherein the user-mode authorization of the process is switched to kernel-mode authorization by the call gate selector via the entry point in the global descriptor table, and is switched back after the operation with kernel-mode authorization has been performed.

17. The method as claimed in claim 16, wherein a far call stated by the call gate selector points to the call gate descriptor, a CPU switches an instruction pointer to the entry point, when a caller from the call gate sends a call, if the caller has kernel-mode authorization.

18. The method as claimed in claim 17, wherein the instruction pointer has kernel-mode authorization, is switched to the entry point, to perform the operation with kernel-mode authorization in the kernel mode, and is switched back to user-mode authorization after performing the operation with kernel-mode authorization.

19. A storage medium for storing a computer program providing a method for performing kernel-mode operations, comprising using a computer to perform the steps of:

providing a kernel-mode generator;

generating a kernel-mode interface using the kernel-mode generator to generate a call gate with performing a kernel-mode operation with kernel-mode authorization in a kernel mode;

providing an authorization interface to connect a user mode to the kernel mode; and

switching a process from the user mode to the kernel mode via the call gate through the authorization interface to perform the kernel-mode operation with kernel-mode authorization.

20. The storage medium as claimed in claim 19, wherein the authorization interface sends a call gate request to the kernel-mode interface generator to generate the kernel-mode interface driver, the call gate is generated according thereto, and the authorization interface directs the process to enter the kernel mode through the call gate.

21. The storage medium as claimed in claim 19, wherein the kernel-mode operation is an operation with Ring 0 authorization level in the kernel mode.

22. The storage medium as claimed in claim 19, wherein the process, a user-mode operation, is capable of user-mode authorization in a protected mode.

23. The storage medium as claimed in claim 22, wherein the user-mode authorization is Ring 3 authorization level in the protected mode.

24. The storage medium as claimed in claim 23, wherein the call gate sets a call gate selector and an entry point in a global descriptor table, having a call gate descriptor and a code-segment descriptor, to perform the operation with kernel-mode authorization in the kernel mode.

25. The storage medium as claimed in claim 24, wherein the user-mode authorization of the process is switched to kernel-mode authorization by the call gate selector via the entry point in the global descriptor table, and is switched back after the operation with kernel-mode authorization has been performed.

26. The storage medium as claimed in claim 25, wherein a far call stated by the call gate selector points to the call gate descriptor, a CPU switches an instruction pointer to the entry point, when a caller from the call gate sends a call, if the caller has kernel-mode authorization.

27. The storage medium as claimed in claim 26, wherein the instruction pointer has kernel-mode authorization, is switched to the entry point, to perform the operation with kernel-mode authorization in the kernel mode, and is switched back to user-mode authorization after performing the operation with kernel-mode authorization.